REMARKS/ARGUMENTS

The claims have been cancelled or amended in view of the Official Action and the cited references. As amended, the independent claims define a method or apparatus for producing a real-time video stream from MPEG encoded video clips from data storage of a video server. Segments of the MPEG encoded video clips are read from the data storage, and decoded by respective first and second decoders in a decoder pair. The real-time video stream is obtained by operating a video switch to switch between a video output of the first decoder and a video output of the second decoder to select a specified In-point frame in the second MPEG encoded video clip that is selectable as any MPEG frame type at any location in an MPEG group of pictures (GOP) structure.

It is respectfully submitted that the independent claims now define one or more of three aspects which distinguish the applicants' invention from the references cited in the Official Action. The first aspect is the video server using configuration commands for configuring the decoders (claims 4, 62 and 70). The second aspect is the decoder sending asynchronous status reports of decoding events to the video server when the decoding events occur (claims 6, 62 and 70). The third aspect is the decoder obtaining MPEG encoded data from the video server by sending a request for data including a decoder data buffer free space value and an offset value indicating any MPEG encoded data previously received from the video server (claims 11, 20, 58).

The video server using configuration commands for configuring the decoders is shown in applicants' FIG. 12, box 121, and FIG. 16 (Configuration) and described in applicants' specification as follows:

> The VideoService function performs configuration discovery by a scan of the data movers connected to a decoder array controller. In other words, the decoder controller is automatically configurable as an external connected device. When VideoService recognizes the unique inquiry string from the decoder array, it sends an appropriate configuration command to each data mover connected to one or more decoder arrays. [Applicants' specification, page 25, lines 13-18.]

> The configuration group allows a data mover to determine the configuration of the decoder array and set up any configuration parameters. Commands in this group include QueryStatus and Configure. [Applicants' specification, page 30, lines 5-7.]

> The Configure command allows a data mover to configure a decoder array that is directly connected to the data mover. Configure() allows the data mover to set up the UDP port to be used for asynchronous status reports and Edit messages, and the operational mode of individual decoders.
>
> /* Mode of an individual decoder */
>
> enum decoderMode_t { DM_UNCONFIGURED, DM_AB_GANGED, DM_SINGLE };
>
> /* Video Standard /*

```
        enum VideoStandard_t {STD_NTSC, STD_PAL};
        /* Type of LTC signal being received */
        enum LTCtype_t { LTC_TYPE25, LTC_TYPE29,
LTC_TYPE30 };
        /* Audio Embedding */
        enum decoderAudioEmbed_t {AE_ENABLED,
AE_DISABLED};
        /* Error Concealment Mode */
        enum ErrorConcealment_t {EC_ FREEZE_FRAME,
EC_BLACK_FRAME };
        /* Configuration of an Individual decoder */
        struct decoderSetup_t {
        int                     decoder;
        unsigned short          flow_control_port;
        decoderMode_t           mode;
        ErrorConcealment_t      error_concealment;
        decoderAudioEmbed_t     audio_embed;
        int audio_delay_ms;     unsigned int
request_interval_ms; };
        /* Configuration of the decoder array */
        struct DecoderArrayConfiguration_t {
        struct in_addr          eth100_IP;
        unsigned short          async_port;
        VideoStandard_t         video_standard;
        LTCtype_t               ltc_type;
        decoderSetup_t          decoders<>;
```

[Applicants' specification, page 32 line 14 to page 33 line 19.]

The decoder sending asynchronous status reports of decoding events to the video server when the decoding events occur, is shown in applicants' FIG. 16 (Asynchronous status reports) and described in applicants' specification as follows:

> The asynchronous status report messages provide asynchronous reports of significant events from a decoder array to a data mover directly connected to the decoder array. Commands in this group include ClipHasStarted, ClipHasEnded, ClipIsEnding, TrapMessage, and EditSummary. [Applicant's specification, page 30, lines 11-14.]

> These reports are sent by the decoder array to the data mover when some significant event has occurred.
>
> ```
> struct ClipActionInfo_t {
> int                 decoder;
> ClipID_t            clip;
> DisplayTime_t       action_time;
> FrameCount_t        frame_count;
> };
> ```
>
> The ClipHasStarted report is issued when a decoder A-B switch occurs. The report is sent by the decoder that is switched on-air and it is sent concurrent with the first frame. The decoder array saves this information so that a subsequent QueryStatus command may recover it. The information is overwritten by the next ClipHasStarted report. The following fields are valid in the ClipActionInfo_t structure:
>
> | | |
> |---|---|
> | decoder d | coder index of decoder brought on-air |
> | clip | ID of clip just started to be displayed |

action_time House Clock time when clip started

frame_count number of frames decoded for the previous clip; when no

previous clip exists, frame_count should be set to '0'

void ClipHasStarted(ClipActionInfo_t);

The ClipHasEnded report is issued when an clip has terminated because it has reached the end of the data stream. It is sent by the decoder that goes off-air and it is sent concurrent with the last frame. A subsequent clip may be pre-rolled for display sometime in the future or with no specified display time. The following fields are valid in the ClipActionInfo_t structure:

decoder decoder index of decoder going off-air

clip ID of clip just ended action_time House Clock time when

clip ended

frame_count number of frames decoded for the clip

void ClipHasEnded(ClipActionInfo_t);

The ClipIsEnding report is issued by the on-air decoder. When the decoder detects the end of data flag in a Data message, it shall send the ClipIsEnding report. It is intended to be used by the data mover to send the on-air decoder a pre-roll command while it is still decoding on-air. By overlapping pre-rolling with streaming, the data mover can play shorter clips. The following fields are valid in the ClipActionInfo_t structure:

decoder decoder index of on-air decoder

clip ID of the clip about to end

void ClipIsEnding(ClipActionInfo_t);

The control station sends the Mark In (MI) and Mark Out (MO) data to the data mover which sends it in turn to the decoder it is

controlling. The EditSummary report is sent by the decoder to the data mover. The data mover will pass the Mark In and Mark Out information to the control station.

```
struct EditSummary_t {
 int                decoder_index;
ClipID_t            clip;
DisplayTime_t       mark_in;
DisplayTime_t       mark_out;
};
void EditSummary(EditSummary_t);
```

The trap message has the following structure:

```
struct TrapMessage_t {
int                 decoder_index;
ClipID_t            clip;
DisplayTime_t       action_time;
FrameCount_t        frame_count;
int                 event_code;
};
```

decoder_index is the decoder index of the decoder initiating the report. A decoder_index value of "-1" shall be used to identify the decoder array as a whole. frame_count is the number of frames decoded from beginning of Clip until action_time. event_code is a unique code that identifies a specified event that crosses a specified threshold. A threshold can be set to "Infinite," i.e., the threshold will never be reached and the event will never be reported. A threshold can be set to "Now," i.e., a single event causes an asynchronous status report. Threshold can be set to any value in between the two extremes.

```
void TrapMessage(TrapMessage_t);
```

[Applicants' specification, page 39 line 15 to page 42 line 12]

The decoder obtaining MPEG encoded data from the video server by sending a request for data including a decoder data buffer free space value and an offset value indicating any MPEG encoded data previously received from the video server, is shown in applicants' FIG. 22, steps 163-166, and described in applicants' specification as follows:

> FIG. 22 shows a flow chart of the program executed by the data mover for the example of FIG. 21. In a first step 161 of FIG. 22, the data mover receives the window (i.e., the decoder data buffer free space) from the decoder. In step 162, the data mover transmits data packets to the decoder to fill the window. In step 163, the data mover receives a new window and offset from the decoder. In step 164, the data mover computes an estimated window size based on the data transmitted to but not received by the decoder by the time of the new window; for example, the estimated window is the sum of the new window and the new offset less the sum of the offset and the data length transmitted since the previous window. In step 165, the data mover transmits more data packets to fill the estimated window. In step 166, execution ends if the end of the video stream for the clip has been reached; otherwise, execution loops back to step 163. [Applicants' specification, page 49, lines 4-14.]

On page 2 of the Official Action, claims 1-3, 5-8, 10, 13-17, 19, 22-25, 52-54, 56, 60-61, and 64-67 were rejected under 35 U.S.C. 102(e) as being anticipated by U.S.

Patent Application Publication 2002/0129374 A1 to Freeman et al. In response, claims 1-3 have been canceled, claim 5 has been amended to depend on claim 6, claim 6 has been re-written in independent form and amended to recite "transmitting asynchronous status reports of decoding events from the decoders to the video server when the decoding events occur," claims 7 and 8 have been amended to depend from claim 6, claims 10 and 13-16 have been amended to depend from claim 6, claim 17 has been canceled, claim 19 has been amended to depend from claim 20, claim 22 has been canceled, claims 23-25 have been amended to depend on claim 20, claim 52 has been canceled, claims 53-54 and 56 have been amended to depend from claim 58, claims 60-61 have been canceled, and claims 64-67 have been amended to depend from claim 62.

Page 4 of the Official Action says:

> Regarding claims 6, 54, 56, Freeman discloses the use of a backchannel encoder 368, which transmits data back to the headend, which includes edit requests (interactive selection) and asynchronous status reports (demographic data for advertising) (paragraphs 118, 120, 169-187).

In response, the applicants' claims referring to the asynchronous status reports have been amended to distinguish Freeman by pointing out that the asynchronous status reports are reports of decoding events from the decoders, and the reports are transmitted to the video server when the decoding events occur. Support for these amendments are

found in applicants' FIG. 16 and in applicants' specification on page 30, lines 11-14, and on page 39 line 15 to page 42 line 12. It is not seen where Freeman discloses that the identified decoders 110A, 110B of FIG. 4 transmit asynchronous status reports of decoding events to a video server when the decoding events occur.

Page 5 of the Official Action says:

> Regarding claims 14, 23, and 66 Freeman discloses that the video server prepares for the switch between the first and second decoders by initiating a stream of MPEG data to the second decoder so that the data from the first clip does not overlap the second clip (figure 4 buffers 164/165 for the respective first and second decoders 100a/b, paragraphs 124-127, the second stream data is buffered prior to the switch).

It is not seen where Freeman discloses "so that the data from the first clip does not overlap the second clip" as defined in applicants' claims. Applicants' claim 14, for example, recites "so that streaming of MPEG encoded data of the first MPEG encoded video clip from the video server to the first decoder is not overlapped with streaming of MPEG encoded data of the second MPEG encoded video clip from the video server to the second decoder." Freeman's "FIG. 4 shows an alternate, dual tuner embodiment for seamless switching between separate video signals. ... This configuration allows the microprocessor 108 to independently select two different individual time-multiplexed video signals on different channels and data streams. ... In this way it is no longer

necessary to have all of the information contained in one RF channel. Instead the information can be found at different frequencies of the RF spectrum and we will still be able to splice among the streams." (Freeman, page 7, paragraph [0087] to page 8, paragraph [0089]." The dual independent channel nature of Freeman's FIG. 4 configuration suggests that that the streaming of data in the first channel would be overlapped with streaming of data in the second channel. This should also be the case where Freeman's dual channel decoders of FIG. 4 were located at a TV broadcast station switching site for switching between separate NTSC channels as mentioned in Freeman page 12 paragraphs [0152] and [0157].

Claims 53 and 54 have been amended to more clearly define that three distinct dedicated links are being claimed. Support for this amendment is shown in applicants' FIG. 1 (data streaming link 42, control command link 48, asynchronous status reports and edit messages link 49) as described in applicants' specification on page 15 line 21 to page 16 line 10:

> As shown in FIG. 1, each decoder in each decoder pair in the decoder array 22 has a data port for a data streaming link 42, 43, 44, 45, 46, 47 to a respective port in the data mover 28 for the pair. This data streaming link, for example, is a bi-directional Ethernet link providing communications via the User Datagram Protocol (UDP). The respective data mover sends data to the decoder and the decoder flow controls the data by using disconnects to control the data rate. The decoder controller 41 has a control port for a bi-directional Ethernet link 48 to the data mover

> 28 using UDP. The controller server 27 sends decoder control requests over the link 31 to the data mover 28 which sends the commands over the link 48 to the decoder controller 41, and the decoder controller returns command responses over the link 48 to the data mover 28. The decoder controller 41 also has an asynchronous port for a bi-directional Ethernet link 49 to the data mover computer 28 using UDP. The decoder controller sends asynchronous status reports and edit messages over the link 49 to the data mover computer 28.

On page 10 of the Official Action, claims 11, 20, 58, and 68 were rejected under 35 U.S.C. 103(a) as being unpatentable over Freeman in view of U.S. Patent 6,124,878 to Adams. Applicants respectfully traverse. It is respectfully submitted that it would not have been obvious to modify Freeman in view of Adams in such a way as to result in the applicants' invention of claims 11, 20, 58, and 68.

Applicants' claim 11, for example, recites: "each decoder obtains MPEG encoded data from the video server by sending a request for data including a decoder data buffer free space value and an offset value indicating any MPEG encoded data previously received from the video server, and the video server responds to the request for data by sending MPEG encoded data sufficient to substantially fill the data buffer free space taking into consideration MPEG encoded data previously sent but not yet received by said each decoder when said each decoder sent the request for data." This is shown in applicants' FIG. 22, steps 163-166, and described in applicants' specification on page 49, lines 4-14, as reproduced above.

Freeman discloses a compressed digital-data seamless video switching system. (Title.) Seamless switching between video signals on different channels is provided (e.g., FIG. 4), as well as seamless switching between video signals that have been multiplexed on the same channels. The various signals which comprise the interactive program can be switched at the head end (e.g., FIG. 9) rather than at the receiver. (Abstract).

With respect to the decoders requesting and obtaining MPEG2 encoded data from the video server, page 6 of the Official Action refers to Freeman paragraphs 169-187. However, these paragraphs do not detail the decoders "requesting" MPEG2 encoded data. For example, paragraph [0176] says: "Distribution from the encoder to the server and the playback systems can be done through a network or by disk or tape." Paragraph [0179] says: "The advertisements can be pushed to the end users ..." Paragraph [0184] says: "Once received at the server, the Group A stream is forwarded to an MPEG transport switch device in the server 550." Paragraph [0186] says: "The transmitter 554 forwards the digital data stream to the remote reception sites, as previously described."

Page 10 of the Official Action says: "Regarding claims 11, 20, 58, and 68,, Freeman discloses that the server maintains the buffers in a substantially full condition (paragraphs 85, 124-125)." However, these paragraphs do not detail a "substantially full" condition or how it might be achieved. Instead, these paragraphs appear to describe filling the buffer with enough information for decoding in accordance with the MPEG standard. For example, paragraph 85 says: "When the compressed digital video is sent to the video decode function it is first stored in memory 160 until there is enough

information buffered to ensure continuous playback of the video stream ..." Paragraph 124 says: " ... The physical buffer size is defined by the MPEG standard, herein incorporated by reference. Enough time must be allowed at the initial onset of decoding to fill up the buffer with I-frame and other data." Paragraph 125 says: "After buffering, the selected video goes through various steps of an MPEG decode process, which utilizes a variable length decode (VLD) preferably ..."

Page 10 of the Official Action further says: "Freeman fails to disclose sending a request for data including a buffer free space value and an offset value indicating any data previously transmitted but not yet received from the server, the server responding to the request by sending data to substantially fill the buffer." Therefore, the Official Action proposes to combine Freeman with the Adams reference.

Page 10 of the Official Action further says: "Adams discloses a system in which a buffer transmits a fullness value to a server and compares the amount of data current within the data, as well as the age of data previously received from a server and remains in the buffer to a server (figure 9, column 10, lines 14-52, figure 10, column 11, lines 5-60), this data is utilized to determine how much data is to be sent to the buffer, to ensure that he buffer does not become too full."

Adams discloses a full service network (FSN) for a cable TV system providing three communication channels that extend between a headend and each set-top within the FSN, namely (1) forward-application transport (FAT) channels that supply data from the headend to all or to only addressed ones of the set-tops,(2) a forward-data-channel (FDC)

that supplies data from the headend to all or to only addressed set-tops, and (3) a reverse-data-channel (RDC) that supplies data from the set-tops to the headend. The FDC carries eight types of traffic, namely (1) conditional access message; (2) entitlement management messages; (3) broadcast data; (4) network management services messages or information; (5) general massaging; (6) application downloading; (7) Internet Protocol external device data services, and (8) VBR downloading. A fixed bandwidth FDC provides a first bandwidth portion for the high priority transmission of items (1), (2) and (3) at a continuous bit rate (CBR). All other items are transmitted over the FDC using at an available bit rate (ABR). A priority system for the selective transmission of these other items is based upon (1) how full a data buffer for an item is, as compared to a fullness reference, (2) how old the oldest data in the data buffer for the item is, as compared to an age reference. The fullness reference and the age reference are usually different for each of these other data items. (Abstract.) An example of VBR downloading is downloading of MPEG-2 compressed data. (See Adams, col. 2, lines 27-28 and col. 2. line 49 to col. 3 line 1.)

In Adams, the buffer in question is one of buffers 93-97 of a multiplexer of the headend as shown in Adams' FIG. 8, and not an MPEG decoder buffer at the set-top. Adams col. 9 line 66 to col. 10 line 7 says:

> A feature of this invention provides that in relation to FDC 26, multiplexer 80 at headend 11 operates to handle the content of an individual buffer 93-97 in accordance with a priority system that is based on (1) the state of fullness of a

buffer 93-97, and the age of the oldest data that is in a buffer 93-97. Stated in another way, priority of handling data that is within a buffer 93-97 is based upon the answer to the two questions; How full is the buffer?, and How old is the first-in-data that is in the buffer?.

The output 98 of demultiplexer 81 comprises a similar plurality of eight output buffers 100-107. Each of these eight individual output buffers 100-107 receives its own individual input from FDC 26, and again provides eight individual outputs designated 40-47, wherein each individual output of demultiplexer 82 carries the information as above defined relative to inputs 40-47 of multiplexer 80.

FIG. 9 is another showing of input 84 of FIG. 8's multiplexer 80 that comprises the five FIFO buffers 93-97. Each of the five buffers 93-97 provides a first output 114, Bf whose signal content comprises the state of fullness of a buffer 93-97, and a second output signal 118, Ba whose signal content comprises the age of the oldest data that is in a buffer 93-97.

Moreover, in Adams FIG. 9, as described in Adams, col. 9, lines 31 to 56:

> When threshold 113 is exceeded by the fullness signal 110 that is provided by a buffer 93-97, that buffer's threshold network 112 generates an output signal 114, i.e. signal Bf. For example, output signal 114, Bf comprises a binary number whose magnitude is indicative of the state of fullness 110 of the associated buffer 93-97. ... Output signal 118 from the threshold networks 116, signal Ba, is a binary number that indicates the number of microseconds

that the oldest data packet of a FIG. 3 data item has resided in the associated buffer 93-97.

In Adams, the signals (Ba) and (Bf) are used for selective transmission of the low priority items over the FDC 26. This may help keep a transmission buffer not too full, but this is different from substantially filling the decoder data buffer.

Therefore, it is respectfully submitted that a head end buffer 90-97 in Adams is not analogous to the decoder buffer in applicants' claims, and the signals (Ba) and (Bf) are not analogous to the decoder data buffer free space value and an offset value indicating any MPEG encoded data previously received from the video server, as recited in applicants' claims. Nor is it seen where Adams discloses "taking into consideration MPEG encoded data previously sent but not yet received by said each decoder when said each decoder sent the request for data."

Page 10 of the Official Action concludes: "Therefore, it would have been obvious to one skilled in the art at the time of the invention to modify Freeman to utilize the buffer monitoring features, including a fullness and age of data attribute of Adams, for ensuring that a buffer does not become too full or too old." However, modification of Freeman in this fashion would not result in applicants' invention. Such a modification may keep transmission buffers in the server from becoming too full or too old, but it would not result in the decoders requesting data from the server, and the server

responding by supplying data to the decoders, in the fashion defined in applicants' claims. Where the prior art references fail to teach a claim limitation, there must be "concrete evidence" in the record to support an obviousness rejection. "Basic knowledge" or "common sense" is insufficient. *In re Zurko*, 258 F.3d 1379, 1385-86, 59 U.S.P.Q.2d 1693, 1697 (Fed. Cir. 2001).

On page 11 of the Official Action, claims 9, 12, 21, 57, 59, and 69 were rejected under 35 U.S.C. 103(a) as being unpatentable over Freeman in view of U.S. Patent 5,742,623 to Nuber. In response, these claims have been cancelled or amended to depend from independent claims that now recite one or more of the three features introduced above that distinguish the references cited in the Official Action.

On page 11 of the Official Action, claim 63 was rejected under 35 U.S.C. 103(a) as being unpatentable over Freeman in view of U.S. Patent 6,441,832 to Tao. In response, claim 63 has been amended to depend from claim 62.

On page 12 of the Official Action, claims 4, 18, 55, 62 and 70-73 were rejected under 35 U.S.C. 103(a) as being unpatentable over Freeman in view of U.S. Patent 6,470,378 to Tracton. In response, claims 4, 18, 55, 62, and 70 have been amended to explicitly define that the configuration commands are used by the video server for configuring the decoders by the video server setting operational modes of the decoders.

(Claims 71, 72, and 73 are dependent upon claim 70.) For example, claim 4 has been amended to recite "the control commands include configuration commands used by the video server for configuring the decoders by the video server obtaining configuration status of the decoders and by the video server setting operational modes of the respective decoders." Support for this amendment is shown in shown in applicants' FIG. 12, box 121, and FIG. 16 (Configuration) and described in applicants' specification on page 25, lines 13-18, page 30, lines 5-7, and page 32 line 14 to page 33 line 19, as reproduced above.

Page 13 of the Official Action says:

> Freeman fails to disclose transmitting configuration commands to allow the video server to determine a configuration of the decoder pair and set up configuration parameters.
>
> Tracton discloses a system in which a decoder is polled for its capabilities and configuration information, after which a version of the video content is transmitted which is appropriate for the device (figures 5/6, column 4, line 33-column 5, lines 46, column 7, line 15-column 8, line 5).

It is respectfully submitted that the amendments to the claim distinguish the proposed Freeman/Tracton combination because the video server transmitting video

content which is appropriate for the decoder in Freeman/Tracton does not suggest the video server configuring the decoders by setting operational modes of the decoders.

Regarding claim 72, it is not seen where Freeman discloses that "data from the first clip does not overlap the second clip." See applicants' remarks above with respect to claims 14, 23, and 66.

On page 14 of the Official Action, claim 74 was rejected under 35 U.S.C. 103(a) as being unpatentable over Freeman in view of Tracton and further in view of Adams. In response, the base claim 70 has been amended, and also it is respectfully submitted that the limitations expressly recited in claim 74 are not disclosed in Freeman/Tracton/Adams. See the applicants' remarks above with respect to claim 11.

On page 15 of the Official Action, claim 75 was rejected under 35 U.S.C. 103(a) as being unpatentable over Freeman in view of Tracton and further in view of Nuber. In response, the base claim 70 has been amended to distinguish Freeman/Tracton/Nuber. See applicant's remarks above with respect to claims 4 and 6.

New claims 76-80 have been added, which are readable on the elected species of figure 2. Support in the original specification for new claims 76-80 is found in applicants' FIG. 16 ("Asynchronous status reports: ... Commands: ClipHasStarted, ClipHasEnded") and as described in applicants' specification on page 40 lines 1-21.

In view of the above, reconsideration is respectfully requested, and early allowance is earnestly solicited.

Respectfully submitted,

22 Feb. 2006

Richard C. Auchterlonie
Reg. No. 30,607

NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, 53rd Floor
Houston, TX 77002
713-751-0655